Aprendizagem 2021/22

Homework IV – Group 072

I. Pen-and-paper

1) E-step

$$\mu_1 = X_1 = \begin{bmatrix}2\\4\end{bmatrix} \qquad \mu_2 = X_2 = \begin{bmatrix}-1\\-4\end{bmatrix} \qquad X_3 = \begin{bmatrix}-1\\2\end{bmatrix} \qquad X_4 = \begin{bmatrix}4\\0\end{bmatrix}$$

$$\Sigma_1 = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix} \qquad \Sigma_2 = \begin{bmatrix}2 & 0\\0 & 2\end{bmatrix}$$

$$\pi_1 = p(c_1 = 1) = 0.7 \quad \pi_2 = p(c_2 = 1) = 0.3 \quad D = 2$$

$$p(X_n \mid c_k = 1) = \mathcal{N}(X_n \mid \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{D/2}} \cdot \frac{1}{|\Sigma_k|^{1/2}} \cdot exp\left(-\frac{1}{2} \cdot (X_n - \mu_k)^T \Sigma_k^{-1} \cdot (X_n - \mu_k)\right)$$

$$p(X_1 \mid c_1 = 1) = \frac{1}{2\pi}$$

$$p(X_2 \mid c_1 = 1) = \frac{1}{2\pi} \cdot exp\left(-\frac{1}{2} \cdot (X_2 - \mu_1)^T \Sigma_1^{-1} \cdot (X_2 - \mu_1)\right) = \frac{1}{2\pi} \cdot exp\left(-\frac{1}{2} \cdot \begin{bmatrix}-3 & -8\end{bmatrix} \cdot \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix} \cdot \begin{bmatrix}-3\\-8\end{bmatrix}\right)$$
$$= \frac{1}{2\pi} \cdot e^{-\frac{73}{2}} = 2.23909 \cdot 10^{-17}$$

$$p(X_3 \mid c_1 = 1) = \frac{1}{2\pi} \cdot \exp\left(-\frac{1}{2} \cdot \begin{bmatrix}-3 & -2\end{bmatrix} \cdot \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix} \cdot \begin{bmatrix}-3\\-2\end{bmatrix}\right) = \frac{1}{2\pi} \cdot e^{-\frac{13}{2}} = 0.00023$$

$$p(X_4 \mid c_1 = 1) = \frac{1}{2\pi} \cdot \exp\left(-\frac{1}{2} \cdot \begin{bmatrix}2 & -4\end{bmatrix} \cdot \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix} \cdot \begin{bmatrix}2\\-4\end{bmatrix}\right) = \frac{1}{2\pi} \cdot e^{-10} = 7.22562 \cdot 10^{-6}$$

$$p(X_1 \mid c_2 = 1) = \frac{1}{2\pi} \cdot \frac{1}{2^{\frac{3}{4}}} \cdot \exp\left(-\frac{1}{2} \cdot \begin{bmatrix}3 & 8\end{bmatrix} \cdot \begin{bmatrix}2 & 0\\0 & 2\end{bmatrix} \cdot \begin{bmatrix}3\\8\end{bmatrix}\right) = \frac{e^{-\frac{73}{4}}}{2\pi \cdot 2^{\frac{3}{4}}} = 1.12247 \cdot 10^{-9}$$

$$p(X_2 \mid c_2 = 1) = 0.09463$$

$$p(X_3 \mid c_2 = 1) = 0.00001$$

$$p(X_4 \mid c_2 = 1) = 3.34603 \cdot 10^{-6}$$

$$p(c_k = 1 \mid X_n) = \pi_k \cdot \mathcal{N}(X_n \mid \mu_k, \Sigma_k) = \pi_k \cdot p(X_n \mid c_k = 1)$$

$$p(c_1 = 1 \mid X_1) = 0.7 \cdot p(X_1 \mid c_1 = 1) = 0.11140$$

$$p(c_1 = 1 \mid X_2) = 1.56736 \cdot 10^{-17}$$

$$p(c_1 = 1 \mid X_3) = 0.00016$$

$$p(c_1 = 1 \mid X_4) = 5.05794 \cdot 10^{-6}$$

$$p(c_2 = 1 \mid X_1) = 3.3674 \cdot 10^{-10}$$

$$p(c_2 = 1 \mid X_2) = 0.028389$$

$$p(c_2 = 1 \mid X_3) = 0.000003$$

$$p(c_2 = 1 \mid X_4) = 1.00381 \cdot 10^{-6}$$

$$\wp(X_n) = \sum_{k=1}^{K} \wp(c_k = 1 \mid X_n) = \sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(X_n \mid \mu_k, \Sigma_k)$$

$\wp(X_1) = 0.11140 + 3.3674 \cdot 10^{-10} \approx 0.11140$

$\wp(X_2) = 1.56736 \cdot 10^{-17} + 0.028389 \approx 0.028389$

$\wp(X_3) = 0.000163$

$\wp(X_4) = 6.06175 \cdot 10^{-6}$

$$\gamma(c_{nk}) = \wp(c_k = 1 \mid X_n) = \frac{\wp(c_k = 1, X_n)}{\wp(X_n)}$$

$\gamma(c_{11}) \approx 1$ $\qquad$ $\gamma(c_{12}) = \dfrac{3.3674 \cdot 10^{-10}}{0.028389} = 1.18616 \cdot 10^{-8}$

$\gamma(c_{21}) = \dfrac{1.56736 \cdot 10^{-17}}{0.11140} = 1.406966 \cdot 10^{-16}$ $\qquad$ $\gamma(c_{22}) \approx 1$

$\gamma(c_{31}) = \dfrac{0.00016}{0.11140} = 0.0014363$ $\qquad$ $\gamma(c_{32}) = \dfrac{0.000003}{0.028389} = 0.00010567$

$\gamma(c_{41}) = \dfrac{5.05794 \cdot 10^{-6}}{0.11140} = 0.00004540$ $\qquad$ $\gamma(c_{42}) = \dfrac{1.00381 \cdot 10^{-6}}{0.028389} = 0.000035359$

M-step

$$N_k = \sum_{k=1}^{K} \gamma(c_{nk})$$

$N_1 = 1.0014817$ $\qquad$ $N_2 = 1.0000920708616$

$$\mu_k = \frac{1}{N_k} \sum_{k=1}^{K} \gamma(c_{nk}) \cdot X_n$$

$$\mu_1 = \frac{1}{1.0014817}\left(1\begin{bmatrix}2\\4\end{bmatrix} + 1.406966 \cdot 10^{-16}\begin{bmatrix}-1\\-4\end{bmatrix} + 0.0014363\begin{bmatrix}-1\\2\end{bmatrix} + 0.00004540\begin{bmatrix}4\\0\end{bmatrix}\right) = \begin{bmatrix}1.99578\\3.99695\end{bmatrix}$$

$$\mu_2 = \frac{1}{1.0000920708616}\left(1.18616 \cdot 10^{-8}\begin{bmatrix}2\\4\end{bmatrix} + 1\begin{bmatrix}-1\\-4\end{bmatrix} + 0.00010567\begin{bmatrix}-1\\2\end{bmatrix} + 0.000035359\begin{bmatrix}4\\0\end{bmatrix}\right) = \begin{bmatrix}-0.99987\\-3.99942\end{bmatrix}$$

$$\Sigma_k = \frac{1}{N_k} \sum_{k=1}^{K} \gamma(c_{nk}) \cdot (X_n - \mu_k) \cdot (X_n - \mu_k)^T$$

$$\Sigma_1 = \begin{bmatrix}0.01307 & 0.00822\\0.00822 & 0.00645\end{bmatrix} \qquad \Sigma_2 = \begin{bmatrix}0.00088 & 0.00070\\0.00070 & 0.00436\end{bmatrix}$$

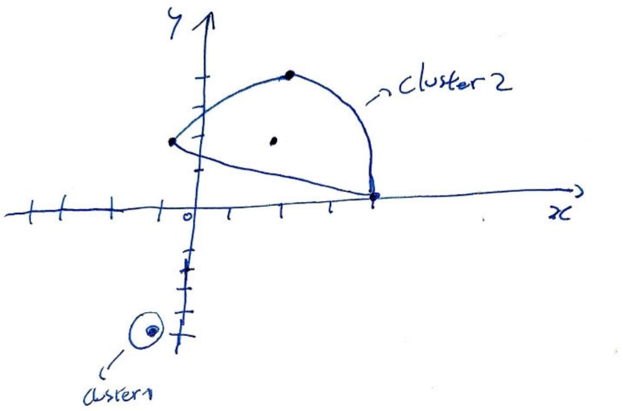
$$\pi_k = \wp(c_k = 1) = \frac{N_k}{N}$$

$\pi_1 = \wp(c_1 = 1) = \dfrac{1.0014817}{4} = 0.250370425$ $\qquad$ $\pi_2 = \wp(c_2 = 1) = \dfrac{1.0000920708616}{4} = 0.2500225$

TÉCNICO LISBOA

Aprendizagem 2021/22

Homework IV – Group 072



2)

The silhouette coeficiente is calculated for a specific instance x_i

Silhouette $x_1 = s(x_1) = 1 - \frac{a}{b}$

Where a, is the average distance of x_i to points in its cluster.

Where b, is the average distance of x_i to points in another cluster.

Assuming that x_3 , x_4 belong to c_1, we obtain the following

$$a = \frac{1}{2}\left[\left\|\begin{bmatrix}2\\4\end{bmatrix}-\begin{bmatrix}-1\\2\end{bmatrix}\right\| + \left\|\begin{bmatrix}2\\4\end{bmatrix}-\begin{bmatrix}4\\0\end{bmatrix}\right\|\right] = \frac{1}{2}\,[3.605551 + 4.472136] = 4.038823$$

$$b = \left[\left\|\begin{bmatrix}2\\4\end{bmatrix}-\begin{bmatrix}-1\\4\end{bmatrix}\right\|\right] = 2$$

$$s(x_1) = 1 - \frac{4.038823}{2}$$

3)

i)

MLP with three hidden layers with as much nodes as the number of input variables.

We have a total of 5 layers, input layer, three hiddens layers and lastly, the output layer.
Assuming the followjng format (5,5,5,5,5).
Between the input layer and first hiddens layer we have:

5 x 5 Matrix , so 25 parameters
5 x 1 Bias Vector

The same occurence repeats to the following:
1st to 2nd hidden layer
2nd to 3rd hiddens layer
3rd to the output layer
The total number of parameters, let's call it T is given by:

$$T = 4*((5*5) + 5) = 120$$

ii)

Decision tree assuming input variables are discretized using three bins.

The initial class dimensionality is 5, nevertheless by applying the three bins quantization, the initial continous values are now transformed/ associated to three different and contiguous intervals, taking d, as the new dimensionality, d = 3.

Since we're working with a decision tree, its vc dimension is given by 2^{d} where d is the dimensionality. Follows that:

$$d_{VC} = 2^{3} = 8$$

iii)

Bayesian classifier with a multivariate Gaussian likelihood

In order to estimate the VC dimension of the aforementioned classifier we first need to calculate its total number of parameters.

The PDF of a multivariate gaussian is given by the following:

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{u})^{T}\Sigma^{-1}(\mathbf{x}-\mathbf{u})}$$

We need to compute the mean vector as well as the covariance matrix.

The mean vector is a 5 x 1 matrix, 5 parameters

The covariance is a 5 x 5 matrix, since it's symmetric we only need to take in account one occurrence of the repeating values, 15 parameters.

We're working with a two-class classifier, follows that the total number of parameters T is the following:

$$T = 2 * (5 + 15) + 1 = 41$$

II. Programming and critical analysis

4)

a) After applying the k-means clustering with k=2 and k=3 to the breast.w.arff we get the ECRS 13.5 and 6.67 which means that error of clustering for k=2 and k=3 is very low.

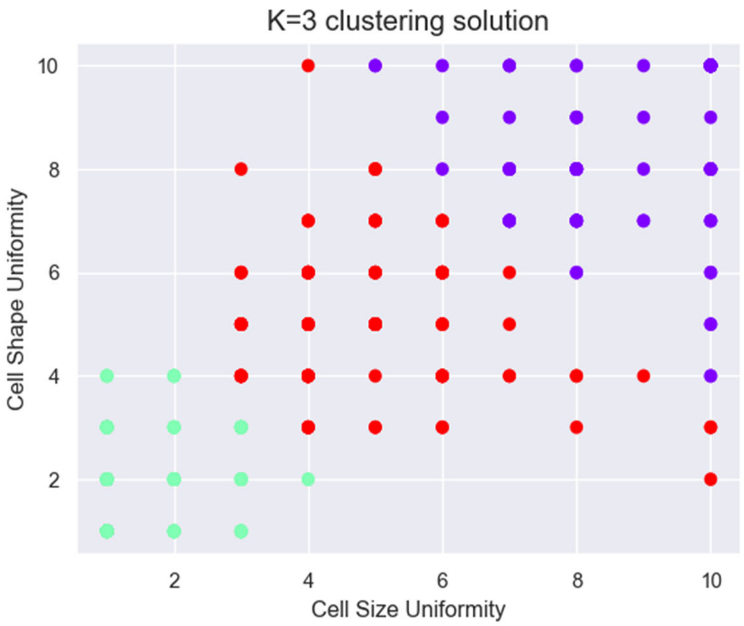
On one hand, this means that both models are well classified, and the clustering of the data is somewhat cohesive and representative due to the errors being somewhat close to 0. On the other hand, we can't assume that the errors are very low as well, or that this is somehow good classified because there isn't any specific upper bound that tells us if the model is badly classified and clustered.

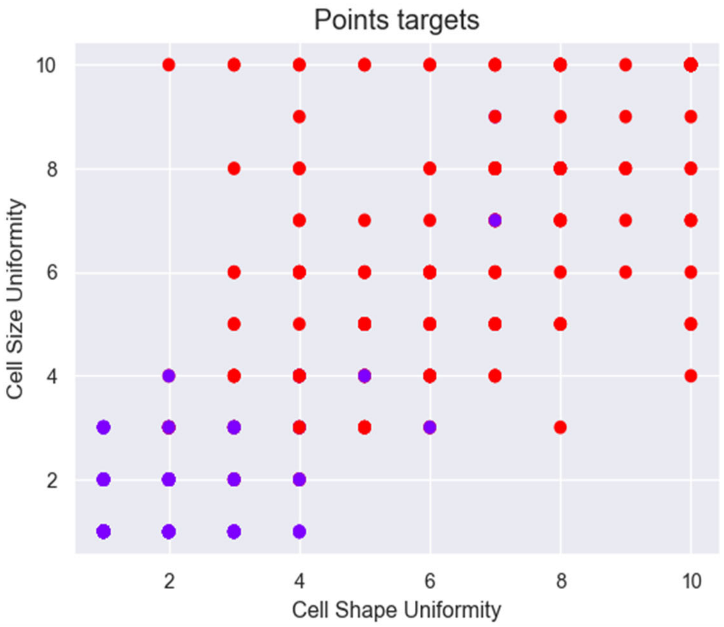
To conclude, since the errors are somewhat closer to 0 and that both are significantly lower than our sample size (683) we can assume that the errors are fairly low for the sample size and that the models are well classified with very few points noise points, which tend to cause the error in the clustering in case they get chosen as centroids.

b) After applying the k-means clustering with k=2 and k=3 to the breast.w.arff we get the silhouette coefficients of 0.5967981179111456 and 0.5256774849851862. With that, we can conclude that the clusters on both models are cohesive (meaning that the objects within the clusters are well matched to its own cluster and poorly matched to neighboring clusters).

Furthermore, since the silhouette coefficient is the maximum value of the mean of the silhouette values of the clusters, we can see that the clustering configuration is appropriate due to both silhouette coefficients being very close to 1.

5)


K=3 clustering solution


Points targets

6) From the empirical data gathered from 5 we can see that the clusters are cohesive despite some noise points in the 2nd and 3rd cluster, we can also see that the produced clustering solution is fairly accurate with the points of each cluster being close from each other (especially in the first cluster).

However, there's some points mainly in the 2nd and 3rd clusters that shouldn't probably be on those clusters due to being closer to other points in other clusters. Moreover, we can see in the targets plot that for example in the 1st cluster that there are points, for example when y=4, that are closer to the 1st cluster than the 2nd, however they're classified in the 2nd cluster.

Furthermore, if we look at both plots (the clusters and targets) we can see the inaccuracies of the clustered solution, with this probably happening due to the choice of the centroids when applying the algorithm to make the clusters, for example, due to the overlapping of some points, it plausible that there were some points where the target was 0 that were in the 2nd and 3rd clusters, overlapping with ones with target 1, that could've been chosen as centroids during any iteration of the algorithm, explaining the inaccuracies shown in the 2nd and 3rd clusters of the clusters plot.

To conclude, the overall quality of the solution is accurate, especially the 1st cluster, despite the inaccuracies and choices used in the algorithm and noise points in the data.

III. APPENDIX

1.

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from sklearn.mixture import GaussianMixture
```

```
X = np.array([[2, 4], [-1, -4], [-1, 2],[4,0]])
medias=X[1:3]
pesos = np.array([0.7,0.3])
cov1 = np.array(np.mat('1 0;0 1'))
cov2 = np.array(np.mat('0.5 0;0 0.5'))
precisoes = np.array([cov1,cov2])

gm = GaussianMixture(n_components=2,
random_state=0,reg_covar=1,means_init=medias,weights_init=pesos,precisions_init=precisoes).fit(X)

labels = gm.predict(X)
frame = pd.DataFrame(X)
frame['cluster'] = labels
frame.columns = ['Weight', 'Height', 'cluster']

color=['blue','green','cyan', 'black']
for val in gm.means_:
     plt.scatter(val[0],val[1])


for k in range(0,3):
    data = frame[frame["cluster"]==k]
    plt.scatter(data["Weight"],data["Height"],c=color[k])


plt.show()
```

4.

```
import numpy as np
import pandas as pd
import statsmodels.api as sm
import matplotlib.pyplot as plt
from sklearn.metrics import silhouette_score
import seaborn as sns
sns.set()
from sklearn.cluster import KMeans

def depth_of_clusters(data):
     t1_c1=0
     t1_c2=0
     t1_c3=0
     t0_c1 = 0
     t0_c2 = 0
     t0_c3 = 0
#Checks if val is of target 1 or 0 (val[0]) and in which cluster they belong (val[1])
     for val in data:
          if val[0] == 1 and val[1] == 0:
               t1_c1+=1
          elif val[0] == 1 and val[1] == 1:
               t1_c2+=1
          elif val[0] == 1 and val[1] == 2:
               t1_c3+=1
          elif val[0] == 0 and val[1] == 0:
               t0_c1+=1
          elif val[0] == 0 and val[1] == 1:
               t0_c2+=1
          elif val[0] == 0 and val[1] == 2:
               t0_c3+=1
     return [(t0_c1,t1_c1),(t0_c2,t1_c2),(t0_c3,t1_c3)]

def Calculates_ECR(lst,k):
     def max(data):
          if data[0] > data[1]:
               return data[0]
          else:
               return data[1]
     if k==2:
          c1_size=lst[0][0]+lst[0][1]
          c2_size = lst[1][0] + lst[1][1]
          ecr = (1/k)*( (c1_size-max(lst[0])) + (c2_size-max(lst[1])) )
          return ecr
     elif k==3:
          c1_size = lst[0][0] + lst[0][1]
          c2_size = lst[1][0] + lst[1][1]
          c3_size = lst[2][0] + lst[2][1]
          ecr = (1 / k) * ((c1_size - max(lst[0])) + (c2_size - max(lst[1])) + (c3_size -
max(lst[2])) )
          return ecr

#seperates training data (lst) from targets(resposta)
f = open("data1.txt", "r")
lst = f.readlines()
resposta = []
for i in range(0, len(lst), 1):
    lst[i] = list(eval(lst[i]))
    resposta += [lst[i][-1]]
    lst[i] = lst[i][:-1]

X = np.array(lst)
y = np.array(resposta)
f.close()

#reads the data from dataframe
data = pd.read_csv('result-breast3.csv')

ndata= data.to_numpy()

#Applies the kmeans to the the data for K=2 and K=3
kmeans2 = KMeans(2)
kmeans3 = KMeans(3)

kmeans2.fit(data)
kmeans3.fit(data)

#Calcultes the silhouette score clusters of the data for K=2 and K=3
score2 = silhouette_score(data,kmeans2.labels_)
score3 = silhouette_score(data,kmeans3.labels_)

print("Silhouette score for k=2: "+str(score2))
print("Silhouette score for k=3: "+str(score3))

#Adds 2 columns, one with the data targets and the other with the cluster
#they belong for K=2 and K=3
data['Class']=y
data_clusters2 = data.copy()
data_clusters3 = data.copy()

data_clusters2['Clusters'] = kmeans2.labels_
data_clusters3['Clusters'] = kmeans3.labels_

#Creates an array with the data targets and clusters
x2 = data_clusters2.iloc[:,9:11].to_numpy()
x3 = data_clusters3.iloc[:,9:11].to_numpy()

#Gets the ammount of points that belong to targets 0 and 1 in each cluster
lst1 = depth_of_clusters(x2)
lst2 = depth_of_clusters(x3)
```

```
#Calculates de ECR for both K=2 and K=3
ECR2=Calculates_ECR(lst1,2)
ECR3=Calculates_ECR(lst2,3)

print("ECR for k=2: "+str(ECR2))
print("ECR for k=3: "+str(ECR3))
```

5.

```
import numpy as np
import pandas as pd
import statsmodels.api as sm
from sklearn.feature_selection import mutual_info_classif as MIC
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
from sklearn.cluster import KMeans

#seperates training data (lst) from targets(resposta)
f = open("data1.txt", "r")
lst = f.readlines()
resposta = []
for i in range(0, len(lst), 1):
    lst[i] = list(eval(lst[i]))
    resposta += [lst[i][-1]]
    lst[i] = lst[i][:-1]

X = np.array(lst)
y = np.array(resposta)
f.close()

#shows an array with the features mutual information
mi_score = MIC(X,y)
print(mi_score)
features =
['Clump_Thickness','Cell_Size_Uniformity','Cell_Shape_Uniformity','Marginal_Adhesion','Single_Epi_Ce
ll_Size','Bare_Nuclei','Bland_Chromatin','Normal_Nucleoli','Mitoses']
top_features=[]
for i in range(0,len(mi_score),1):
    top_features += [(mi_score[i],i,features[i])]

top_features = sorted(top_features, key=lambda tup: tup[0])[-2:]
print("Top 2 features: " + top_features[0][2]+" and "+top_features[1][2])

#creates a dataframe with the data
data = pd.read_csv('result-breast3.csv')

#creates a dataframe with the top-2 features with higher mutual information
x = data.iloc[:,top_features[1][1]:top_features[0][1]+1]

#performs k-means clustering and creates a column in the original dataframe
#with the cluster index
kmeans = KMeans(3)
kmeans.fit(x)
identified_clusters = kmeans.fit_predict(x)
data_with_clusters = data.copy()
data_with_clusters['Clusters'] = identified_clusters

#shows the plot with the clusters
plt.scatter(data_with_clusters[top_features[0][2]],data_with_clusters[top_features[1][2]],c=data_wit
h_clusters['Clusters'],cmap='rainbow')
plt.xlabel(top_features[0][2].replace("_", " "))
plt.ylabel(top_features[1][2].replace("_", " "))
plt.title('K=3 clustering solution', fontdict={'fontsize':15})
plt.show()

#Adds the targets column in the data
data_with_targets = data.copy()
data_with_targets['Class'] = y

#shows the plot with every point's target
plt.scatter(data_with_targets[top_features[0][2]],data_with_targets[top_features[1][2]],c=data_with_
targets['Class'],cmap='rainbow')
plt.xlabel(top_features[0][2].replace("_", " "))
plt.ylabel(top_features[1][2].replace("_", " "))
plt.title('Points targets', fontdict={'fontsize':15})
plt.show()
```

END